Ekstraksi Informasi pada Makalah Ilmiah dengan Pendekatan *Supervised Learning*

Information Extraction on Scientific Papers with Supervised Learning Approach

Aditya Iftikar Riaddy[1], Yuliant Sibaroni, S.Si, M.T.[2], Annisa Aditsania, S.Si, M.Si[3]

[123] Program Studi Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

[1]airiaddy@live.com [2]ysibaroni@gmail.com [3]annisaaditsania@gmail.com

ABSTRAK

Makalah ilmiah merupakan laporan hasil penelitian yang dipublikasikan, dan seringkali dijadikan referensi untuk mengembangkan penelitian lainnya. Semakin banyaknya makalah ilmiah yang tersedia secara *online* memicu kebutuhan akan informasi tentang makalah tersebut, terutama untuk mesin pencari. Untuk mendapatkan informasi dari makalah ilmiah yang jumlahnya banyak dengan cepat dan akurat, dibutuhkan suatu sistem ekstraksi informasi otomatis pada makalah ilmiah. Salah satu pendekatan yang dapat dilakukan untuk melakukan ekstraksi informasi adalah *supervised learning*.

Dalam penelitian ini dilakukan ekstraksi informasi pada makalah ilmiah dengan pendekatan *supervised learning*. Hasil dari penelitian ini didapatkan kombinasi fitur dan *classifier* yang terbaik untuk mengekstraksi setiap informasi dari makalah ilmiah.

Kata kunci: ekstraksi informasi, supervised learning, makalah ilmiah, natural language processing

ABSTRACT

Scientific papers are published research report, and often cited to develop other researches. More scientific papers available online triggers the need for information extraction of the papers, especially for search engine. To retrieve information from large number of papers quickly and accurately, a system of automatic information extraction on scientific paper is needed. One of the approach that can be done for information extraction is supervised learning.

In this research, information extraction from scientific paper with supervised learning is done. The results of this research are best combinations of feature and classifier to extract every information from scientific paper.

***Keywords**: information extraction, supervised learning, scientific journal, natural language processing*

1. Pendahuluan

Seiring dengan berkembangnya internet, ketersediaan data secara *online* juga terus meningkat, terutama data yang berbentuk teks. Data teks tersebut banyak mengandung informasi. Akan tetapi, informasi-informasi yang ada pada teks tersebut seringkali tersembunyi, karena bentuk dari teks tersebut yang tidak terstruktur. Oleh karena itu dibutuhkan ekstraksi informasi, yaitu suatu sistem untuk mencari data spesifik dalam *natural language text* [10]. Sebagai contoh adalah berita tentang bencana alam. Dalam berita tersebut terkandung beberapa informasi seperti jenis bencana yang terjadi, lokasi, dan jumlah korban. Namun seluruh informasi tersebut tersembunyi dalam bentuk kalimat-kalimat, sehingga untuk dapat mengambil informasi yang penting tersebut keseluruhan teks harus dianalisa. Jika berita tentang bencana tersebut jumlahnya sangat banyak, tentu akan merepotkan jika seluruh berita harus dianalisa. Dari melimpahnya data tersebut dan informasi yang dikandungnya, timbul kebutuhan untuk mengekstraksi informasi secara otomatis dari data yang berupa teks [11].

Data teks yang turut mengalami perkembangan adalah data yang berbentuk jurnal penelitian. Ekstraksi informasi pada jurnal penelitian dari *header* dan referensi jurnal sangat dibutuhkan untuk berbagai aplikasi, misalnya pencarian berbasis bidang, analisis penulis, dan analisis kutipan [12]. Munculnya berbagai mesin pencari untuk jurnal penelitian juga turut meningkatkan urgensi akan ekstraksi informasi dari jurnal penelitian. Oleh karena itu dibutuhkan suatu metode yang dapat mengekstraksi informasi-informasi yang terkandung dalam *header* dan referensi jurnal penelitian.

Supervised learning merupakan salah satu metode *machine learning* yang bertujuan untuk memetakan suatu *input* ke dalam sebuah *output* yang nilainya sudah disediakan [1]. Penelitian tentang ekstraksi informasi dengan *supervised learning* sebelumnya pernah dilakukan di bidang finansial [9], biologi [5], dan halaman HTML [4]. Pada penelitian ini akan dilakukan ekstraksi informasi pada makalah ilmiah dengan pendekatan *supervised learning*.

Dalam penelitian yang dilakukan oleh Peng [11], penggunaan kombinasi fitur yang berbeda-beda dapat mempengaruhi performa *classifier*. Begitu juga dengan penggunaan *classifier* yang berbeda. Oleh karena itu, dalam penelitian ini dilakukan pencarian kombinasi fitur dan *classifier* yang terbaik untuk setiap informasi.

2. Ekstraksi Informasi

Ekstraksi informasi merupakan suatu sistem untuk mencari data spesifik dalam *natural language text*. Data yang akan diekstraksi biasanya didapatkan dari sebuah *template* berupa formulir atau tabel yang akan diisi dengan kalimat-kalimat atau komponen-komponen dari teks tersebut [10]. Informasi yang akan diekstraksi dari makalah ilmiah pada penelitian ini ada 6, yaitu:

- judul,
- nama penulis,
- afiliasi,
- alamat,
- *email*, dan
- *website*.

Alur sistem ekstraksi informasi yang dilakukan pada penelitian ini dapat dilihat pada gambar 1.

3. Makalah ilmiah

Makalah ilmiah atau *scientific paper* adalah laporan yang ditulis dan dipublikasikan untuk mendeskripsikan hasil penelitian orisinal [6]. Bagian dari makalah yang akan diekstraksi informasinya adalah bagian kepala dari makalah. Bagian kepala dari makalah yang didefinisikan adalah dari judul hingga informasi penulis dan afiliasi.

Makalah yang digunakan sebagai data dalam penelitian ini didapatkan dari *website* ACL Anthology Reference Corpus (acl-arc.comp.nu.edu.sg). Jumlah keseluruhan makalah yang digunakan adalah 1027 makalah. Makalah-makalah tersebut kemudian dibagi menjadi dua dengan rasio 9:1 untuk *training* dan *testing*.

4. Ekstraksi Fitur

Kelompok fitur yang akan digunakan pada tugas akhir ini adalah fitur lokal dan tata letak dari penelitian Peng [11] dan *named-entity* dari penelitian Finkel [FINKEL]. Fitur lokal merupakan karakteristik yang terdapat dalam *string* setiap baris kalimat. Fitur tata letak merupakan posisi suatu baris kalimat dalam bagian kepala makalah. Kemudian fitur *named entity* adalah fitur non-lokal yang diekstraksi dengan *library* Stanford Named Entity Recognition (NER). Fitur-fitur yang digunakan secara lengkap dapat dilihat di tabel 1.

Tabel 1 Daftar fitur yang digunakan

Nama fitur	Deskripsi
Fitur Lokal	
INITCAPS	Dimulai dengan huruf kapital
ALLCAPS	Seluruh karakter adalah huruf kapital
CONTAINSDIGIT	Mengandung digit angka
ALLDIGITS	Seluruh karakter adalah digit angka
PHONEORZIP	Nomor telepon atau kode pos
CONTAINSDOTS	Mengandung paling sedikit satu titik
CONTAINSDASH	Mengandung paling sedikit satu garis
ACRO	Akronim / singkatan
LONELYINITIAL	Inisial seperti A.

SINGLECHAR	Hanya mengandung satu karakter
CAPLETTER	Hanya mengandung satu huruf kapital
PUNC	Tanda baca
URL	Alamat URL
EMAIL	Alamat *email*
Fitur Tata Letak	
LINE_START	Berada di awal baris
LINE_IN	Berada di pertengahan baris
LINE_END	Berada di akhir baris
Fitur *Named Entity*	
LOCATION	Nama lokasi
PERSON	Nama orang
ORGANIZATION	Nama organisasi
MONEY	Nominal uang
PERCENT	Nominal dalam persen
DATE	Waktu dalam penanggalan
TIME	Waktu jam

Kelompok-kelompok fitur tersebut kemudian dikombinasikan untuk dicari kombinasi fitur terbaik pada masing-masing informasi yang dapat diekstraksi. Kombinasi fitur yang digunakan dalam penelitian ini ada empat, yaitu:

- fitur lokal,
- fitur lokal + tata letak,
- fitur lokal + *named entity*, dan
- fitur lokal + tata letak + *named entity*

5. *Supervised Learning*

Pada [7]], dijelaskan bahwa s*upervised learning* merupakan salah satu metode untuk mengklasifikasikan masing-masing objek dalam data ke beberapa kelas. Pada *supervised learning* setiap objek pada suatu data memiliki fitur, yaitu ciri-ciri yang ada pada masing-masing objek. Setiap objek dalam suatu data memiliki jumlah fitur yang sama. Fitur digunakan sebagai input untuk menentukan kelas pada objek. Dalam *supervised learning*, kelas dari masing-masing objek sudah diketahui. Oleh karena itu, permasalahan yang dihadapi dalam *supervised learning* adalah bagaimana memetakan objek ke dalam kelas yang tepat menggunakan fitur-fitur yang dimiliki oleh setiap objek.

Klasifikasi pada *supervised learning* dilakukan dengan melakukan *training* (pelatihan) untuk membentuk model. *Classifier* (algoritma klasifikasi) akan membentuk model yang beradaptasi sesuai dengan fitur-fitur yang ada pada data. Model yang dihasilkan dapat berupa *tree*, *rule*, atau suatu fungsi yang dapat memprediksikan suatu kelas berdasarkan fitur-fitur yang dimiliki oleh data tersebut.

Langkah selanjutnya adalah melakukan validasi terhadap model yang dihasilkan. Metode validasi yang digunakan dalam penelitian ini adalah *k-fold cross-validation*. Data dipartisi menjadi *k* bagian sama rata. Setelah itu secara bergantian 1 partisi menjadi data *testing* dan *k*-1 partisi lainnya menjadi data *training*. Nilai *k* yang digunakan pada penelitian ini adalah 10.

Pada tugas akhir ini algoritma supervised learning yang akan digunakan ada tiga, yaitu *Support Vector Machine* (SVM), *Naïve Bayes*, dan *Random Forest*. Implementasi dari ketiga *classifier* tersebut pada penelitian ini menggunakan *library* dari Wakaito Environment for Knowledge Analysis (WEKA).

5.1. Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu algoritma machine learning yang paling populer. Prinsip kerja dari SVM adalah memisahkan dua buah kelas yang terpisah secara linier dengan membuat sebuah garis pemisah yang disebut *hyperplane* [1]. Dalam SVM dikenal istilah margin, yaitu jarak antara garis *hyperplane* dengan data yang paling dekat yang disebut dengan *support vector*.

Hyperplane yang paling optimal adalah yang memiliki margin terbesar. Dengan margin yang besar, maka kita dapat menghindari kesalahan dalam pengklasifikasian data. Contohnya, jika terdapat sebuah data baru pada kelas "+" yang posisinya berada sedikit lebih dekat ke kelas "•", maka data tersebut masih dapat diklasifikasikan ke kelas yang tepat.

Jika kita memiliki suatu data $\mathbb{X} = \{\}$ dengan 2 kelas berbeda, yaitu ω dan ω, di mana $= +1$ jika $\in \omega$ dan $= -1$ jika $\in \omega$, persamaan garis *hyperplane* dapat dinyatakan sebagai berikut:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{5.1}$$

di mana w dan b adalah parameter dari model. Kemudian untuk menguji kelas data *testing*, kita dapat menggunakan rumus berikut:

$$f(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b) \tag{5.2}$$

jika $f(\mathbf{x}) = 1$ pilih ω, atau jika $f(\mathbf{x}) = -1$ pilih ω.

5.2. Naïve Bayes

Naive bayes merupakan salah satu metode klasifikasi yang berakar pada teorema Bayes. Pada Naive Bayes, setiap atribut dalam data dianggap independen antara satu dan lainnya [8]. Pada teorema Bayes, jika terdapat dua kejadian yang terpisah (misal A, dan B) maka teorema Bayes dapat dirumuskan sebagai berikut:

$$P(A \mid B) = \frac{P(A)}{P(B)} P(B \mid A) \tag{5.3}$$

Keterangan:

$P(A \mid B)$: Peluang kejadian A bila B terjadi

$P(A)$: Peluang kejadian A

$P(B)$: Peluang kejadian B

$P(B \mid A)$: Peluang kejadian B bila A terjadi.

Teorema Bayes sering juga dikembangkan karena berlakunya hukum probabilitas total menjadi seperti berikut:

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{\sum_{i=1}^{n} P(A_i \mid B)} \tag{5.4}$$

Keterangan:

$P(A \mid B)$: Peluang kejadian A bila B terjadi

$P(B)$: Peluang kejadian B

$P(B \mid A)$: Peluang kejadian B bila A terjadi.

$P(A_i \mid B)$: Peluang kejadian A ke-i bila B terjadi, untuk i=1...n

Untuk menjelaskan bahwa teorema naive bayes, perlu diketahui bahwa proses klasifikasi memerlukan petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis maka teorema diatas dapat disesuaikan menjadi:

$$P(C \mid F_1 \ldots . F_n) = \frac{P(C)P(F_1 \ldots . F_n \mid C)}{P(F_1 \ldots . F_n)} \tag{5.5}$$

Variabel ℓ merepresentasikan kelas, sementara $F_1 \ldots . F_n$ merepresentasikan petunjuk yang akan digunakan untuk melakukan klasifikasi. Rumus tersebut menjelaskan peluang masuknya sampel dengan karakteristik tertentu.

Untuk mencari kelas data yang kita uji, maka kita akan membandingkan posterior (nilai peluang suatu sampel berada di kelas) untuk masing masing kelas. Nilai posterior dapat diperoleh dengan cara berikut:

$$C_{NB} = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(f_i \mid c) \quad (5.6)$$

Dengan c adalah variabel kelas yang tergabung pada suatu himpunan kelas ℓ [NATALIUS].

5.3. *Random Forest*

Random Forest merupakan *classifier* bertipe *decision tree* yang diperkenalkan oleh Leo Breiman. Prinsip kerja dari *random forest* adalah membentuk banyak *classifier tree* dengan fitur-fitur yang dipilih secara acak dari dataset. Kumpulan *classifier tree* tersebut kemudian digunakan untuk mengklasifikasikan vektor input. [3].

Random Forest menggunakan metode *bagging* (*bootstrap aggregating*), yang artinya menggunakan beberapa model yang dibentuk dari sampel data, kemudian dikombinasikan dengan cara voting. Sampel dipilih secara acak dengan penggantian, artinya beberapa data bisa masuk ke dalam lebih dari satu sampel, dan ada data yang tidak dimasukkan ke dalam sampel (*out of bag*). Metode bagging ini cocok digunakan untuk *classifier* dengan bias yang kecil tetapi memiliki variansi yang besar seperti *decision tree* [2].

Langkah-langkah dalam membangun sistem random forest adalah sebagai berikut [4]:

1. Bangun beberapa sampel data dengan penggantian dari data asli. Sampel-sampel data tersebut memiliki ukuran yang sama dan masing-masing digunakan untuk membangun tree yang berbeda.
2. Pilih sebanyak variabel secara acak dari jumlah variabel input, M secara independen untuk setiap node. Jika tidak ditentukan, maka secara default nilai ini adalah pembulatan ke bawah dari rumus berikut: (5.7)
3. Kembangkan setiap tree hingga tidak ada node lagi yang bisa dihasilkan. Tahap ini sama dengan tahap training pada decision tree.
4. Untuk melakukan klasifikasi atau testing, klasifikasikan data menggunakan seluruh tree. Kemudian lakukan voting untuk menentukan kelas pada data tersebut.

6. Pengukuran Kinerja

A. Word Accuracy

Word Accuracy adalah perhitungan performasi dengan menghitung persentase jumlah kata yang diklasifikasikan dengan benar dengan jumlah seluruh klasifikasi. *Word Accuracy* dapat dihitung dengan rumus:

(6.1)

Keterangan:
A: Jumlah *true positive*
B: Jumlah *false negative*
C: Jumlah *false positive*
D: Jumlah *true negative*

B. F1-measure

F1-measure adalah pengukuran kinerja dengan mengukur *precision* dan *recall*. *Precision* dan *recall* dapat didefinisikan sebagai berikut:

(6.2)

(6.3)

Kemudian F1 dapat dihitung dengan rumus berikut:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6.4)$$

7. Testing Model

Testing (pengujian) dilakukan sebagai validasi pada model *classifier* dan kombinasi fitur terbaik untuk setiap informasi. Validasi ini diperlukan untuk menguji apakah performa dari model yang sudah dibentuk tetap konsisten jika digunakan pada data baru. Hasil dari pengujian model ini kemudian diukur juga kinerjanya dengan *word accuracy* dan *F1-measure*.

8. Hasil Penelitian

Hasil dari penelitian ini adalah daftar informasi dan kombinasi kelompok fitur dan classifier yang memiliki performa paling besar. Rangkuman hasil dari eksperimen yang telah dilakukan dapat dilihat pada tabel 2.

Tabel 2 Rangkuman hasil percobaan

Informasi	Fitur	*Classifier*	*Training*		*Testing*	
			Word Accuracy	*F1-Measure*	*Word Accuracy*	*F1-Measure*
title	lokal + tata letak + *named entity*	*Random Forest*	97.207	0.908	97.861	0.935
author	lokal + tata letak + *named entity*	SVM	97.061	0.906	97.995	0.939
affiliation	lokal + tata letak + *named entity*	*Random Forest*	96.505	0.948	96.390	0.946
address	lokal + tata letak + *named entity*	SVM	99.259	0.975	99.733	0.990
email	lokal + *named entity*	*Random Forest*	99.947	0.998	99.866	0.996
	lokal + tata letak + *named entity*	*Random Forest*				
website	lokal	*Naïve Bayes*, SVM	99.987	0.889	100.000	1.000
	lokal + tata letak	SVM, *Random Forest*				
	lokal + *named entity*	SVM, *Random Forest*				
	lokal + tata letak + *named entity*	SVM				

9. Analisis

Penggunaan seluruh kelompok fitur merupakan alternatif kombinasi fitur yang terbaik untuk seluruh informasi. Hal ini dirasa wajar, karena semakin banyak karakteristik yang digunakan tentu *classifier* akan semakin mudah untuk mengenali ciri-ciri dari setiap kelas informasi yang ada.

Setiap kelompok fitur memiliki pengaruh yang berbeda-beda terhadap informasi yang diekstraksi, tergantung dengan karakteristik dari informasi yang berkaitan. Informasi seperti judul makalah tentu akan mendapatkan keuntungan dari fitur tata letak, karena posisi judul yang hampir selalu berada di awal baris. Kemudian pada informasi nama pengarang dan afiliasi, penambahan fitur *named entity* dapat meningkatkan performa *classifier* karena fitur tersebut dapat mengekstrak entitas bernama seperti nama orang dan nama organisasi. Informasi *email* juga mendapat keuntungan dari fitur *named entity* karena banyak didapati alamat email yang mengandung nama orang. Dan informasi *website* dapat diekstraksi dengan akurat walaupun hanya menggunakan fitur lokal, karena pola informasi tersebut yang sudah jelas sehingga mudah untuk diekstraksi tanpa membutuhkan banyak atribut.

10. Kesimpulan

Dari penelitian ini, penggunaan fitur yang tepat dapat meningkatkan akurasi *classifier supervised learning*. Karena setiap informasi memiliki karakteristik yang berbeda, maka diperlukan fitur yang berbeda-beda untuk mengekstraksi

informasi tersebut dengan lebih akurat. Selain itu, semakin banyak fitur yang digunakan juga semakin baik, karena akan semakin banyak ciri-ciri yang diperhitungkan pada saat melatih model *classifier*.

11. Saran

Saran yang dapat diberikan dari hasil penelitian ini adalah pada tahap *preprocessing* dapat dikembangkan dengan melakukan pelabelan otomatis. Kemudian mengeksplorasi fitur-fitur yang baru yang dapat digunakan untuk meningkatkan kinerja dari ekstraksi informasi ini.

Daftar Pustaka

[1] Alpaydin, E., 2010, *Introduction to Machine Learning*, Massachusetts Institute of Technology.

[2] Breiman, L., 1994, *Bagging Predictors*, Department of Statistics, University of California.

[3] Breiman, L., 2001, *Random Forests*, Kluwer Academic Publishers

[4] Changuel, S., Labroche, N., Bouchon-Meunier, B., 2009, *A General Learning Method for Automatic Title Extraction from HTML Pages*, Laboratoire d'Informatique de Paris 6.

[5] Craven, M., Kumlien, J., 1999, *Constructing Biological Knowledge Based by Extracting Information from Text Sources*, American Association for Artificial Intelligence.

[6] Day, R. A., "*What Is a Scientific Paper*?", [online], Available: http://mason.gmu.edu/~jjohnsto/Dayarticle.htm, [Diakses 17 Maret 2015].

[7] Dougherty, G., 2013, *Pattern Recognition and Classification*, Springer.

[8] Lowd, D., Domingos, P., *Naive Bayes Models for Probability*, Department of Computer Science and Engineering, University of Washington, Seattle.

[9] Malik, H. H., Bhardwaj, V.S., Fiorletta, H., 2011, *Accurate Information Extraction for Quantitative Financial Events*, Thomson Reuters.

[10] Nahm, Y. U., *Text Mining with Information Extraction*.

[11] Peng, F., McCallum, A., 2004, *Accurate Information Extraction from Research Paper using Conditional Random Fields*, Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL).

[12] Soderland, S., 1999, *Learning Information Extraction Rules for Semi-Structured and Free Text*, Netherlands, Kluwer Academic Publishers.